

Aleksi Kariniemi

OpenStackin hallinnointi

Apache LibCloudin välityksellä



Tradenomi

Tietojenkäsittely

Syksy 2015



KAJAANIN
AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

TIIVISTELMÄ

Tekijä: Kariniemi Aleksi

Työn nimi: OpenStackin hallinnointi Apache LibCloudin välityksellä

Tutkintonimike: Tradenomi, tietojenkäsittely

Avainsanat: Python, virtualisointi, IaaS, pilvipalvelu, OpenStack, LibCloud

Tämä työ on Kainuun Edun tilaama selvitys Apache LibCloud käskykirjaston hyödyllisyydestä OpenStack-virtualisointiohjelmiston käyttöä varten, ja dokumentaatio kyseisen käskykirjaston asennusprosessista ja käytöstä. FORGE Service Development Lab on DIGILEn hallinnoima palvelu, jossa yritykset voivat kehittää palveluitaan ennen kuin ne julkaistaan kuluttajien käytettäväksi. Kainuun Etu vuorostaan hoitaa teknisen tuen osaa FORGEssa, ja haluavat tietää, miten Apache LibCloud käskykirjasto toimii FORGE:n yhteydessä.

FORGE perustuu avoimen lähdekoodin OpenStack-virtualisointiohjelmistoon. Apache LibCloud käskykirjasto asentuisi OpenStack-ohjelmiston yläpuolelle omaksi komentokerrokseen, ja käskykirjastossa on mahdollisuus ohjata myös muita virtualisointiohjelmia saman käyttöliittymän kautta. FORGE kuitenkin käyttää nykyhetkellä vain yhtä ohjelmistoa, OpenStackia, joten työ keskittyi selvittämään edellä mainittujen kahden ohjelmistojen yhteistoiminnan.

Testaus suoritettiin tekemällä asennus LibCloudista ja käymällä läpi kaikki OpenStackin kanssa yhteensopivat komennot. Ainoastaan kommentojen lopputulokset kirjattiin ylös, eikä käskykirjaston ohjelmakoodia tarkkailtu testauksen aikana.

Testauksen tulokset paljastivat vakavia puutteita LibCloudin käskyjen toiminnassa. OpenStackin käyttöä ajatellen tärkeimmät komennot palauttivat virheilmoituksia, tai ilmoittivat olevansa keskeneräisiä. Täten LibCloud ei ole kunnossa jossa sitä voisi suositella käytettäväksi. Lisäksi FORGE on lakkautettu, joten kehitysehdotuksia ei voida esittää.

Tämä opinnäytetyö käy samalla asennusohjeesta ja käyttöohjeesta LibCloudille.

ABSTRACT

Author: Kariniemi Aleks

Title of the Publication: Management of OpenStack through Apache LibCloud

Degree Title: Bachelor's Degree of Business Administration, Information Technology

Keywords: Python, virtualization, IaaS, cloud computing, OpenStack, LibCloud

The purpose of the thesis was to determine whether Apache LibCloud -command library could provide substantial utility to the operation of FORGE and OpenStack virtualization software, i.e. the core of the service. The thesis was commissioned by Kainuun Etu as their share of the workload consists of providing user support and helpdesk service to the FORGE users, and they thus have an interest in seeing how Apache LibCloud works with the system.

FORGE Service Development Lab is a service managed by DIGILE, designed to offer a platform for companies to develop their Internet services before public release. Apache LibCloud is planned to be installed on top of OpenStack, on its own command layer, where it could manage multiple virtualization software suites through its interface. FORGE, however, consists only of OpenStack, wherefore a further objective was to focus on investigating how well Apache LibCloud and OpenStack would work together.

Testing was done by performing an installation of Apache LibCloud, and all the commands that were available for OpenStack were tested. Only end results visible to the user were measured, no deep inspection of the code's working and potential flaws was done.

The results revealed glaring flaws in the operations of the installed software. Several commands that were key factors to the use of OpenStack were incomplete and unusable or they returned errors beyond an average user's capabilities to correct. Given that LibCloud is not in a state where it can provide any utility, and FORGE has been shut down, no development recommendations can be made.

None the less the thesis result will serve as a documentation of the installation process and a user manual to the Apache LibCloud version that was available at the time of the task.

SISÄLLYS

Symboliluettelo	4
1 JOHDANTO.....	7
2 Teoria, käsitteet ja historia.....	Virhe. Kirjanmerkkiä ei ole määritetty.
2.1 Virtualisoinnin historia	8
2.2 Virtualisointitekniikoita	9
2.3 Python	11
2.4 Pilvipalvelut	12
2.5 OpenStack	14
3 Tarve, suunnitelma ja toteutus	Virhe. Kirjanmerkkiä ei ole määritetty.
4 Ohjelmisto ja sen rajoitteet	Virhe. Kirjanmerkkiä ei ole määritetty.
5 Ohjelmiston asennus ja käyttö.....	Virhe. Kirjanmerkkiä ei ole määritetty.
6 Ohjelmakäskyt.....	Virhe. Kirjanmerkkiä ei ole määritetty.
6.1 Tietoja hakevat käskyt	23
6.2 Toiminnalliset käskyt	25
6.3 Esimerkkejä komennoista osana skriptejä	30
6.4 Virheet ja havaitut puutteet.....	32
7 SELVITYKSEN YHTEENVETO.....	Virhe. Kirjanmerkkiä ei ole määritetty.
Lähdeluettelo.....	36

SYMBOLILUETTELO

Apache = Vapaan lähdekoodin ohjelmistoja kehittävä, tukeva ja ohjaava säätiö.

API = Vieraskielinen lyhenne termistä 'Application Programming Interface'. Tarkoittaa ohjelmille annettuja määritelmä ja sääntöjä joilla viestitään toisten ohjelmien tai tietokoneen osien kanssa.

Autentikaatio = Prosessi, jossa käyttäjä ja hänen antamansa tiedot varmennetaan oikeiksi, jotta käyttäjä pääsee käyttämään järjestelmää.

CDN = Vieraskielinen lyhenne termistä 'Content Delivery Network'. Tarkoittaa järjestelmiä, jotka hoitavat Internetissä tapahtuvia tiedostojen siirtoja käyttäjille ja järjestelmistä toisiin älykkäästi niin, että siirtojen laatu ei kärsisi.

Dict = Vieraskielinen lyhenne sanasta 'dictionary'. Konekielinen termi, joka tarkoittaa luetteloa.

DIGILE = Suomalainen hanke, joka kehittää Suomen elinkeinoelämän digitalisaatiota.

Digitalisaatio = Valmiiksi olemassa olevien ihmisten käyttöön tarkoitettujen palveluiden siirtäminen Internetiin niin, että niitä voi käyttää sieltä käsin.

Distribuutio = Yleisessä jaossa oleva kokoelma eri ohjelmia, jotka muodostavat oman versionsa Linux-pohjaisesta käyttöjärjestelmästä.

FORGE Service Development Lab = Suomalainen hanke ja pilvipalvelu, joka tarjoaa yrityksille resurssit ja työkalut kehittää omia Internet-pohjaisia palveluita.

Flavour = OpenStackin käyttämä termi. Kertoo, kuinka paljon virtuaalipalvelimelle on annettu eri resursseja käytettäväksi, tai toisin sanoen palvelimen koko.

Hypervisor = Termi ohjelmistoille, jotka hoitavat varsinaisen virtualisoinnin. Luovat ja hallitsevat virtuaalikoneita ja niille jaettavia resursseja.

Infrastructure as a Service, IaaS = Käsitemalli, jossa yritys kokoaa valmiit palvelintilat ja niiden ylläpitoon tarvittavan infrastruktuurin, ja vuokraa asiakkaille tästä valmiista kokoonpanosta tilaa heidän omiin käyttötarkoituksiin.

Käsky/komentokirjasto = Ohjelma, joka vastaanottaa käskyjä ja tulkitsee ne toisille ohjelmille helpottaen useiden ohjelmien yhtäaikaisen käytön.

Käyttöliittymä = Ohjelmistoissa ja laitteissa oleva osa, jonka kautta sitä voidaan käyttää ja komentaa suorittamaan operaatioita.

LibCloud = Virtualisointiohjelmistojen käyttöä varten kehitetty käskykirjasto.

Linux = Linus Torvaldsin kehittämän käyttöjärjestelmätimen nimi.

Linux-pohjainen = Käyttöjärjestelmä, joka käyttää osanaan Linux-käyttöjärjestelmäydintä.

Nova = OpenStack-ohjelmiston komponentti tai palvelu vastuussa virtuaalikoneiden hallinnasta ja pyöryksestä.

OpenStack = Avoimeen lähdekoodiin perustuva virtualisointiohjelmisto.

Pooli = Vieraskielisestä termistä 'pool' johdettu sana. Tarkoittaa yleensä kokoelmaa tietokoneresursseja, jotka eivät ole käytössä ja ovat siten vapaita.

Pilvipalvelu = Internet-palvelu, jossa asiakkaille vuokrataan virtuaalipalvelimia ja -tietokoneita käytettäväksi Internet-yhteyden välityksellä.

Python = Ohjelmointikieli.

Python-Pip = Linux-pohjaisille käyttöjärjestelmille tehty ohjelma, joka asentaa Python-ohjelmointikieleen liittyvät kirjastot.

PyPy = Vaihtoehtoinen ohjelma Python-ohjelmointikielen suorittamiseen.

Rauta, tietokonerauta = Viittaus englanninkieliseen termiin 'hardware'. Tarkoittaa tietokoneen tai palvelimen fyysisiä komponentteja tai kokonaisuudessa tietokonetta tai palvelinta.

Skripti = Tiedosto, joka sisältää joukon komentoja tietokoneelle suoritettavaksi. Voi sisältää myös komentoja eri ohjelmointikielillä.

Snapshot = Vieraskielinen termi, joka tarkoittaa väliaikaista varmuuskopiota virtuaalikoneesta ja kaikista sen sisällä pyörivistä ohjelmista.

Str = Vieraskielinen lyhenne sanasta 'string'. On tietokonetermi, joka tarkoittaa tekstiä joka ei sisällä mitään tietokonemuuttujia tai muita koodi-ilmaisuja.

Sääntöryhmä = Käännös OpenStackin termistä 'security group'. On ryhmä johon voidaan sijoittaa virtuaalikoneita. Sääntöryhmää määrittävät myös turvallisuussäännöt, jotka sitten koskevat ryhmässä olevia virtuaalikoneita.

Terminaali = Tekstipohjainen käyttöliittymä tietokoneeseen.

Toimintaympäristö = Olosuhteet, joissa työ tehdään. Koostuu fyysisestä laitteistosta ja niihin asennetuista ohjelmatyökaluista.

Trusty Tahr = Erään Ubuntu-käyttöjärjestelmän version koodinimi.

Turvallisuussääntö = Käännös OpenStackin termistä 'security group rule'. Tarkoittaa sääntöryhmän sääntöä, joka määrää, mistä Internet-osoitteista ja millä keinoilla voi ottaa yhteyden virtuaalikoneisiin.

Ubuntu = Iso-Britanniassa kehitetty avoimen lähdekoodin Linux-pohjainen käyttöjärjestelmä.

Varmuuskopiointi = Olemassa olevista tiedoista kopion ottaminen varalle, jos alkuperäinen tieto häviää tai turmeltuu.

Virtualisointi = Prosessi, jossa fyysisen tietokoneen tai palvelimen sisällä pyöritetään ohjelmistoa, joka matkii fyysisen tietokoneen tai palvelimen toimintaa ja ominaisuuksia.

1 JOHDANTO

Kainuun Etu OY on Kajaanissa toimistoaan pitävä yritys, jonka päämääränä on kehittää Kainuun maakunnan liiketoimintaa ja yrityksiä. Yritys on julkisella rahoituksella toimiva eikä tavoittele voittoa. Kainuun Edun harjoittama kehitystoiminta auttaa paikallisia yrityksiä luomaan kontakteja, kehittämään niiden liiketoimintoja ja houkuttelemaan sijoituksia kohdealueelleen. Esimerkiksi Nordic Digital Business Summit-tapahtumat ovat heidän järjestämiä. Kajaanin ammattikorkeakoulun opiskelijoille yritys tarjoaa mahdollisuuksia verkostoitua työelämän kontaktien kanssa ja hankkia kokemusta työharjoittelujen ja opinnäytetöiden kautta.

Olin jo aiemmin suorittanut työharjoittelun Kainuun Edulla asiakaspalvelijana DIGILEN FORGE-palvelulle, luoden ja hallinnoiden käyttäjätilejä OpenStack-ympäristössä. Tämän opinnäytteen aihe löytyi samasta aihepiiristä kuin harjoittelu ja pyyntö selvityksestä tuli Kainuun Edun vakiotyöntekijöiltä, jotka olivat kiinnostuneita selvittämään pystyikö heidän OpenStackiin liittyviä työtehtäviä helpottamaan.

Kainuun Edun jäseniltä tuli ensimmäisenä ehdotus selvittää oliko Apache LibCloudista, tässä työssä testatusta ohjelmistosta, varteenotettavaksi vaihtoehdoksi. Heiltä itseltään ei riittänyt työtunteja asiaan perehtymiseen, joten asia vietiin eteenpäin opinnäytetyöksi perusteellisen selvityksen aikaansaamiseksi.

2 TEORIA, KÄSITTEET JA HISTORIA

Tämä työ käsittelee ja työskentelee useiden eri informaatioteknologioiden konseptien kanssa kuten virtualisoinnin tai pilvipalveluiden, jotka voivat olla asiaan perehtymättömille haastavia ymmärtää. Tämän luvun tarkoitus on hälventää tätä hämmennystä ja selittää nämä käsitteet auki.

2.1 Virtualisoinnin historia

Virtualisointi voi vaikuttaa tuoreelta teknologialta, joka on noussut vasta 2000-luvulla yleiseen tietouteen VMware-yhtiön tekemän työn ansiosta, mutta se oikeasti juontaa alkunsa 1960-luvulle suurten keskusyksikkökoneiden ajalle. Silloin vielä tietokoneilla saattoi olla vain yksi käyttäjä, joka saattoi antaa komentoja ja operaatioita keskusyksikön suoritettavaksi. Käyttöliittymät olivat alkeellisia jos niitä edes oli suunniteltu koneille, ja interaktiivisuus oli minimissä. Käskyt ajettiin yhtenä isona ryppäänä tietokoneisiin, ja tietokone tulosti sitten tulokset tai virheilmoitukset ajon lopussa takaisin käyttäjälle (Conroy, 2011).

Muutos käynnistyi vuonna 1963 Massachusettsin teknologiainstituutin tarjouspyynnöstä kehittää tietokonejärjestelmä, joka olisi useamman käyttäjän käytettävissä yhtä aikaa. Kehitystyö päättyi General Electricin yhtiölle, IBM:n sivuutettua mahdollisuuden, jota he pitivät turhana. Sitten kun Bell-yhtiön laboratoriot alkoivat pyytämään samanlaista järjestelmää, tietokonejätti IBM havahtui muutoksen uusiin tuuliin ja tarjoutui kehittämään itse samanlaisen järjestelmän kuin teknologiainstituutti oli pyytänyt. (Conroy, 2011)

IBM:n kehittämät CP/CMS-järjestelmät muodostivat yhden ensimmäisistä virtualisointijärjestelmistä. Keskustietokoneisiin asentunut CP-ohjelmisto oli vastuussa virtuaalisien tietokoneiden luomisesta ja tietokoneresurssien jaosta niille. Käyttöpäätteisiin asennettu kevyt CMS-käyttöjärjestelmä välitti käyttäjien ja ohjelmien käskyt keskustietokoneelle ja takaisin. Jos käyttäjä teki virheen, niin se rajoittui vain käyttäjän omaan virtuaaliseen tietokoneeseen eikä kaatanut koko järjestelmää. (Conroy, 2011)

General Electrics yhtiön kehittämästä järjestelmästä tuli MultiCS, josta myöhemmin jalostuisi Bellin laboratorioden jatkokehitystyön seurauksena UNIX-käyttöjärjestelmä, jonka toimintaperiaatteita tänään lukuisat Linux-johdannaiset käyttöjärjestelmät pyrkivät imitoimaan. (Conroy, 2011)

2.2 Virtualisointitekniikoita

Virtualisointi käytännössä on olemassa olevien tietokoneressurssien kuten laskutehon ja kovalevytilan kokoamista yhteen, ja sitten niiden piilottamista niitä käyttäviltä tahoilta. Hypervisoriksi kutsuttu ohjelmisto hallinnoi virtuaalikoneita ja niille jaettavia resursseja. Hukkaan menevät voimavarat voidaan käyttää paremmin ja vikasietoisuutta lisätä rikkoutumisien varalta. (Red Hat Corporation, 2015)

Työpöytävirtualisoinnissa kokonaiset tietokoneet tehdään virtuaalisiksi. Ne käyttäytyvät kuin fyysiset tietokoneet ja imitoivat niiden käyttäytymistä, mutta ovat oikeasti keskustietokoneissa. Käyttäjät näkevät omalta näytöltään vain niihin välitetyn tavallisen tietokoneen näkymän. (Red Hat Corporation, 2015)

Tietokoneiden virtualisoinnissa ei ole pelkästään yhtä metodia, vaan sen voi toteuttaa useilla eri teknisillä tavoilla. Täydellisessä laitteiston emuloinnissa virtuaalikone käyttäytyy kuin se olisi fyysinen, eikä se tiedosta olevansa virtuaalinen. Virtualisointiohjelmisto tarjoaa virtuaalikoneelle fyysistä laitteistoa mallintavat rajapinnat, joista pyytää laskutehoa ja muistia. Tällä metodilla voidaan emuloida useita eri arkkitehtuureilla suunniteltuja tietokoneita ja asentaa useita käyttöjärjestelmiä ilman erityisempiä valmisteluja. (Red Hat Corporation, 2015)

Paravirtualisoinnissa käyttöjärjestelmiä on muokattu niin, että ne tietävät olevansa virtualisoituja. Tällöin fyysistä laitteistoa emuloivat rajapinnat eivät ole tarpeellisia ja virtuaalikoneet osaavat pyytää resursseja suoraan virtualisointiohjelmistosta. Tämä metodi on kevyempi suorittaa, mutta vaatii käyttöjärjestelmien muokkausta etukäteen ja siten pääsyä niiden lähdekoodeihin. Tästä syystä paravirtualisoinnissa on vähäisempi määrä käyttöjärjestelmiä saatavilla. (Red Hat Corporation, 2015)

'Single Kernel Image' metodissa luodaan kopioita virtuaalikoneiden yllä olevan hypervisor-ohjelmiston käytössä olevasta käyttöjärjestelmästä. Tässä mallissa saatavilla on vain yksi käyttöjärjestelmä ja siitäkin vain yksi tietty versio. Tällä tavalla voidaan tehdä kevyitä virtualisointeja ohjelmistoilla, jotka vaativat tietyn toimintaympäristön. Haittapuolena SKI-metodi voi altistaa virtualisointiohjelmiston haittaohjelmille, jos sitä ei ole suojattu oikein tunkeilijoilta. (Red Hat Corporation, 2015)

Tallennustilan virtualisoinnissa sama resurssien piilottaminen tehdään kovalevyille. Kovalevyt, joiden koko, teknologia ja valmistaja voivat erota toisistaan, järjestellään yhdeksi isoksi yksiköksi, joka sitten jakaa tallennustilaa siltä pyytävälle ohjelmille ja käyttäjille. Lisäksi tallennettuja tietoja voidaan varmuuskopioida, jakaa ja palauttaa helpommin. Vikasietoisuus myös paranee kun tieto on tallennettu useampaan eri sijaintiin levyrikkojen varalta. (Bunn;Simpson;Peglar;& Nagle, 2003)

Verkon virtualisoinnissa piilottamisen kohteena ovat verkkolaitteisto, kytkimet ja verkkoportit. Niiden päälle luodaan joukko virtuaaliverkkoja, jotka voivat käyttää samoja verkkoportteja keskenään samaan aikaan toisistaan tietämättä. Verkkojen virtualisoinnilla voidaan pidentää vanhempien verkkolaitteiden käyttöikää ottamalla ne osaksi uutta kokonaisuutta, mutta verkkolaitteiden läpi virtaavan tiedon maksiminopeus on yhä rajoitteena. (SDxCentral, 2015)

Virtualisointia voi lisäksi toteuttaa pienemmillä osa-alueilla tietokoneiden sisäpuolella. Ohjelmistoja voidaan myös virtualisoida niin, että ne toimivat käyttöjärjestelmästä ja alustasta riippumattomana. Tästä yksi esimerkki on Java-ohjelmointikielellä tehdyt ohjelmat, jotka kootaan käytettävään kuntoon koodista vasta käyttäjän koneella. Ainoa vaatimus ohjelman toimivuudelle on Java-työympäristön asennus koneelle, joka pyörittää Java-ohjelmia. (Conroy, 2011)

2.3 Python

Python-ohjelmointikielen ensimmäinen versio ilmestyi vuonna 1991 Guido van Rossumin kehittämänä (Rossum, Python History, 2009). Sen perusominaisuuksiin kuuluvat mm. dynaamiset muuttujat, joiden tyyppiä ei tarvitse määritellä luontivaiheessa, ja sisennyksien käyttö sulkeiden sijasta funktioiden erottelussa. Muuten kieli jakaa useita funktioiden avainsanoja C-kielen kanssa (Rossum, Introduction and overview, 2009).

Nykyisiltä versioiltaan Python-kieli on kykenevä useisiin käyttökohteisiin, perusohjelmoinnista tieteellisiin laskuihin, graafisien käyttöliittymien kehitykseen ja verkkosovelluksen tekoon kuten OpenStackin (Python Software Foundation, 2015). Esimerkiksi Google käyttää Python-kieltä hakukoneensa yhteydessä, osana botteja, jotka selaavat läpi Internet-sivuja (Rossum, Introduction and overview, 2009).

Muiden kielten rinnalla testattuna Python-kielen suoritusaika asettaa sen nopeampien ohjelmointikielten joukkoon ja sen vaatimat resurssit ovat vähäisiä, mutta ei ilman sen omia erikoisuuksia, kuten eroavaa syntaksia jotka voivat aiheuttaa päänvaivaa (OnlyJob, 2012).

Tämän vuoden marraskuun TIOBE-indeksin mukaan Python olisi viidenneksi suosituin ohjelmointikieli tällä hetkellä. Indeksillä perustuu hakukoneiden ja asiantuntijoiden suorittamiin laskuihin, jotka mittaavat kunkin kielen esiintymistä tuloksissa. (TIOBE Software BV, 2015)

2.4 Pilvipalvelut

Digitaalisation myötä on alkanut ilmestyä palveluita ja palveluntarjoajia, jotka pyrkivät poistamaan tietokoneita haluavilta asiakkaalta pohjatyon tarpeen. Palveluntarjoaja tekee kovan työn säilytystilojen, sähkön, laitteiston ja ohjelmistojen hankinnan osalta ja muotoilee näistä erilaisia palvelupaketteja, joissa asiakkaan työosuus on tämän valittavissa. (Interoute Communications Limited, 2015)

Näin toimivat yritykset, jotka mainostavat 'as a Service' palveluita. Tietokonetoitaa varten ei ole tarve aina perustaa omaa palvelinsalia taikka hankkia ohjelmistoja, vaan joku voi jo myydä niitä palveluna Internetin ylitse. Riittää vain, että asiakas voi päästä käsiksi tarjottuihin tuotteisiin Internetin kautta. (Interoute Communications Limited, 2015)

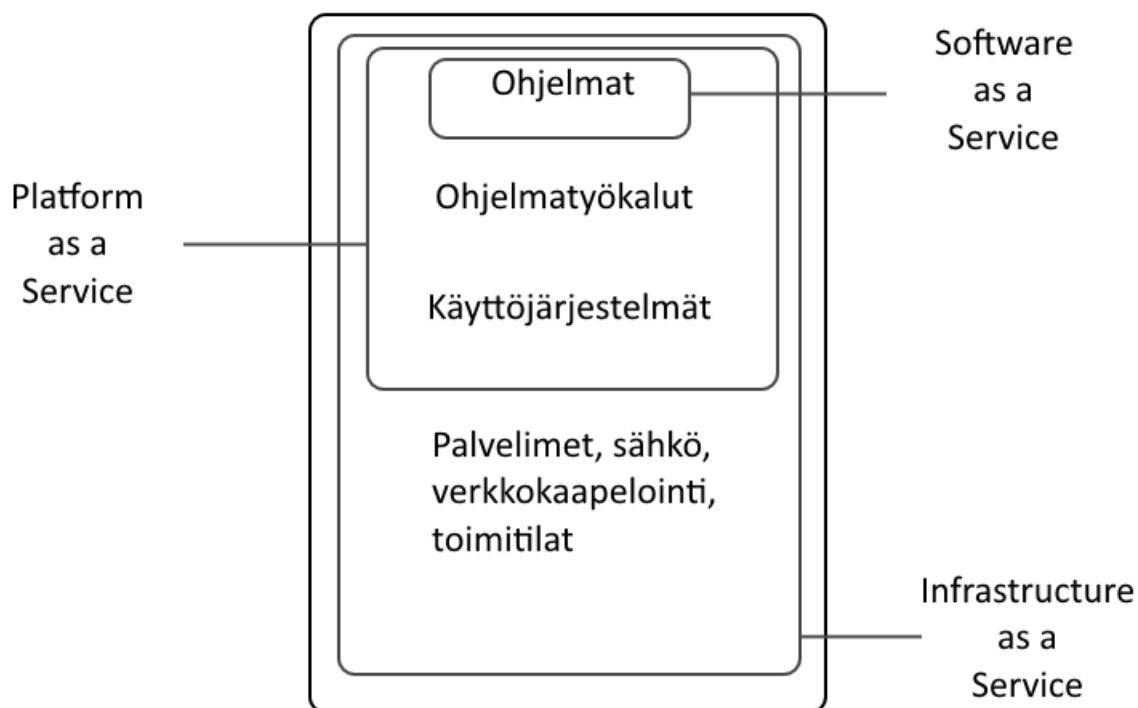
'Infrastructure as a Service' mallissa vuokrataan periaatteessa tilaa palvelinsalista asiakkaalle. Pystytyskulut, fyysisen laitteiston huolto ja järjestely, verkotus, sähkölaskut ja fyysisien tilojen hankinta ja vartiointi jää palveluntarjoajan vastuulle. Asiakkaalle tarjotaan maksusta avaimet käteen sovitulle määrälle tietokoneresursseja, joille voidaan perustaa virtuaalipalvelimia. Näitä palvelinkeskuksesta saatavia voimavaroja voidaan sitten käyttää esimerkiksi asiakkaan omien palveluiden kahdentamiseksi tai varmentamiseksi käyttökatkosten varalta. Kyseessä voi olla myös työympäristön siirto verkkoon etätyön mahdollistamiseksi tai ylimääräisen tallennustilan hankinta. Esimerkkejä tällaisista palveluista ovat Google Cloud Platform ja FORGE Service Development Lab. (Interoute Communications Limited, 2015)

'Platform as a Service' mallissa tarjotaan työkalut ja kehitysympäristö ohjelmistokehittäjille, säästäten asiakkailta lisenssien ja ohjelmistojen hankinnan kanssa kuluva aikaa. Asiakkaat voivat rakentaa haluamansa paketin valitsemalla haluamansa ominaisuudet, ja olla maksamatta ominaisuuksista, joita he eivät halua. Kehitysympäristö voi sisältää vain pelkät työkalut itse ohjelmistojen laatimiseen, tai niissä voi olla muita tukevia ohjelmistoja tallennustilan, versiohallinnan ja tietokantoja varten. Vaihtoehtojen määrä voi olla suppeampi kuin 'IaaS'-palvelussa, mutta 'PaaS'-palveluiden asiakkailla on

enemmän kohdennettu palvelutarve. Esimerkkejä tällaisista palveluista ovat Google App Engine ja Amazon Web Services Elastic Beanstalk. (Interoute Communications Limited, 2015)

'Software as a Service' mallissa rajataan tarjontaa pelkästään Internetin kautta käytettäviin ohjelmiin, joita tarjotaan asiakkaan käytettäväksi. Kuluttajapuolella sovellukset ovat helposti saatavilla useille alustoille kaikkialta missä on Internet-yhteys. Yritysten ja organisaatioiden IT-osastojen ei tarvitse huolehtia itse uuden ohjelmiston käytöstä ja huollosta, vaan vastuu on palveluntarjoajalla. Esimerkkejä ohjelmistoista palveluina ovat Googlen toimisto-ohjelmien kokoelma ja Microsoftin Office 365, tiedostojen jakopalvelu Dropbox ja yhteisöpalvelu Facebook. (Interoute Communications Limited, 2015)

Nämä kolme palvelua ovat hierarkkinen kokonaisuus, joista on rajattu haluttu ominaisuus asiakkaalle myytäväksi paketiksi heidän tarpeen mukaan alla olevan kuvan mukaisesti.



Kuva 1. Havainnekuva erilaisten pilvipalvelumallien lähtökohdista. Mitä korkeammalta valittu palvelu on hierarkiasta, sitä suppeampi tarjottu palvelu on.

Pilvipalvelu-termi on yläkäsite näille kolmelle ja muille samankaltaisille palvelutyypeille. Yhteistä näillä kaikilla on, että kyseiseen palveluun pääsee käsiksi Internet-yhteyden kautta sallien käytön kaikkialta mistä saa yhteyden. Nämä pilvipalvelut kykenevät reagoimaan asiakasmäärien ja kulutuksen vaihteluihin lisäämällä ja leikkaamalla palvelimia, kaistaa ja tilaa. Laskutusta voidaan hoitaa kulutuksen mukaan sallien joustoa myös hinnoissa. Ne voivat tukea olemassa olevia IT-palveluita, joita organisaatiot omistavat, tai muodostaa kokonaan uusia palveluita. (Interoute Communications Limited, 2015)

Ulkopuoliselta taholta ostettua pilvipalvelua kutsutaan julkiseksi pilveksi, koska siellä sijaitseva tieto ja toiminnot pyörivät laitteistolla, jota käyttäjät eivät omista. Yrityksille arkaluontoisen ja tärkeän tiedon säilytys niiden toimitilojen ulkopuolella ei välttämättä sovi, jolloin parempana vaihtoehtona voi olla oman talon sisäisen pilvipalvelun tai yksityisen pilven rakentaminen. Yksityisen pilven rakentamisessa tosin luovutaan ulkopuolisen tahon tekemän työn edusta, sillä yritys ottaa vastuulleen laitteiston ja ohjelmistojen asentamisen ja huollon. Etuna kuitenkin on parempi tietoturvallisuus. Näiden kahden välimaastossa on hybridipilvi, jossa on sekoitettu oman ja ulkopuolisten palveluntarjoajien pilvipalvelut käytettävyyden parantamiseksi. Hybridipilvissä täytyy olla erityisen tarkkana, miten rajoittaa ja määrittää julkisen pilven pääsy yksityiselle puolelle. (Interoute Communications Limited, 2015)

2.5 OpenStack

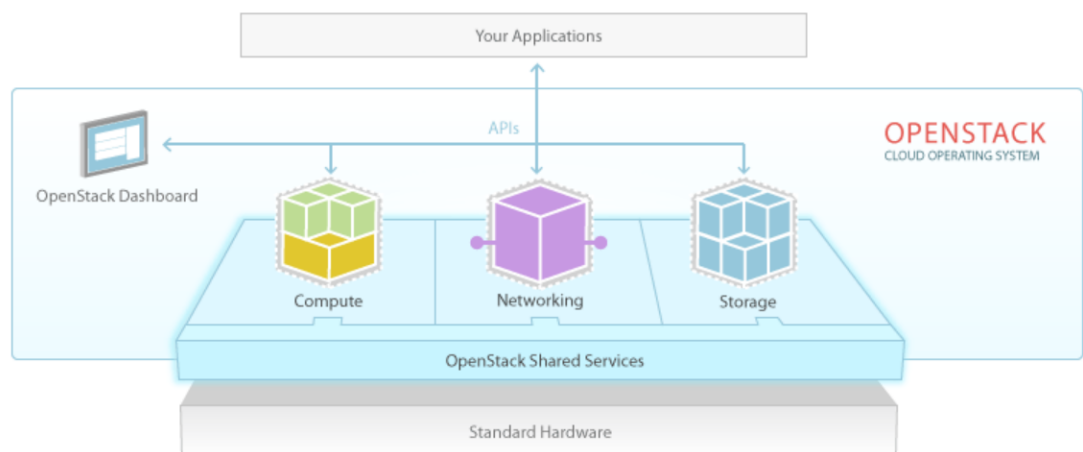
OpenStack on suurille mittakaavoille käytettäväksi suunniteltu virtualisointiohjelmisto, tai toisin sanoen alusta pilvipalveluille. Se koostuu eri tehtäviä varten ohjelmoiduista moduuleista, jotka hoitavat tehtäviä virtuaalikoneiden luomisesta tallennustilan ja verkkoyhteyksien hallintaan ja moniin muihin tarpeisiin.

OpenStack sai alkunsa vuonna 2010 Rackspace-yhtiön ja Yhdysvaltain avaruusvirasto NASAn yhteistyöstä kehittää pilvipalvelu, joka osaisi hyödyntää keskivertokuluttajan käytettävissä olevaa tietokonerautaa. Rackspacella oli valmiina virtuaalikoneita luova hypervisor-moduuli, ja NASAlla vuorostaan oli

omiin tarkoituksiin kehitetty tallennustiloja virtualisoinut moduuli. Rackspacen yrityksessä päätettiin avaruusviraston suostumuksella paketoita kummatkin ohjelmat uudeksi kokonaisuudeksi, joka julkaistaisiin Apache 2.0 käyttölisenssin alaisena avoimena lähdekoodina pilvipalvelualan eduksi. (Moorman, 2010)

Vuonna 2012 ohjelmiston kehitystä valvomaan perustettiin säätiö, ja siitä eteenpäin yhä useampi yritys on ruvennut käyttämään OpenStackia ja tukemaan sen kehitystyötä (BusinessWire, 2012). Vuodelta 2014 oleva tutkimus yksityisien pilvien rakennukseen käytetyistä ohjelmistoista paljasti OpenStackin nauttivan kolmanneksi suosituimman ohjelmiston statusta, heti alan jätin VMwaren ja uuden tulokkaan Microsoft Azuren jälkeen (Marko, 2014). Voidaan siis todeta, että avoimen lähdekoodin ratkaisu oli leviämisen kannalta oikea Rackspacelle.

Kuten muutkin virtualisointiohjelmistot, OpenStack noudattaa samaa abstraktion kaavaa omissa toimintaperiaatteissaan piilottamalla resurssit ja jakamalla niitä käyttäjille tarpeen mukaan. Alla oleva kuva esittelee tämän kaavan.



Kuva 2. Demonstraatio OpenStack-ohjelmiston järjestelmien hierarkiasta, kuten nähty OpenStackin kotisivuilla. OpenStackin palvelut asentuvat omaksi kerroksekseen käytettyyn laitteistoon, jonka ylle virtuaalikoneet ja niissä pyörivät ohjelmistot sitten sijoittuvat.

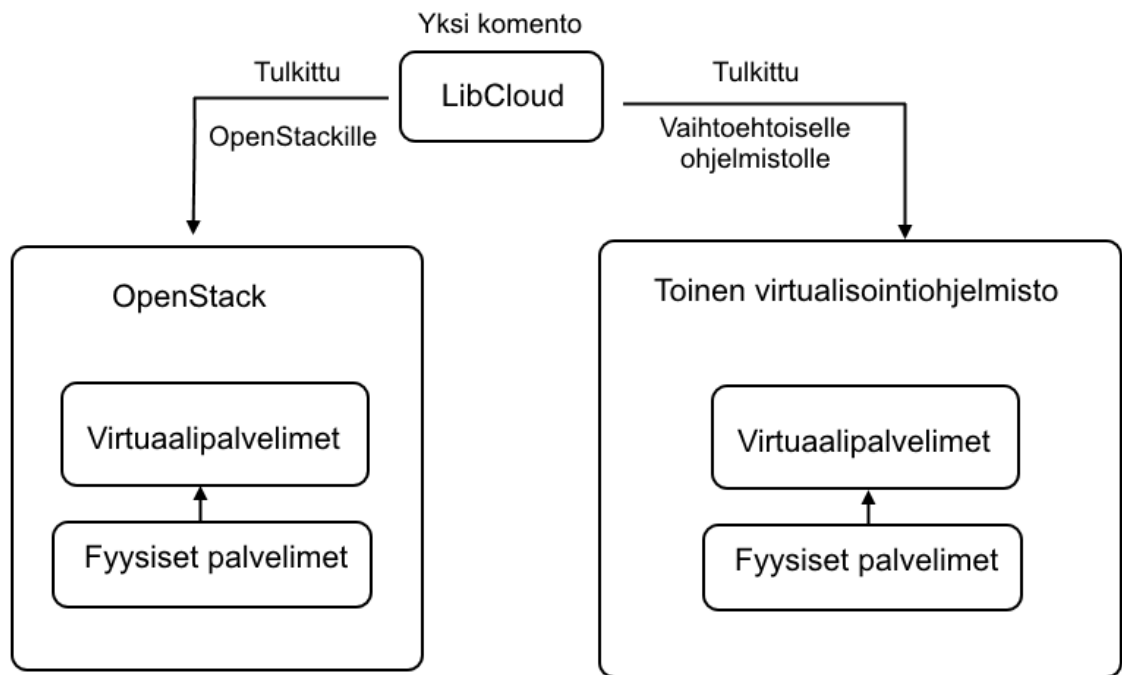
3 TARVE, SUUNNITELMA JA TOTEUTUS

Kainuun Etu, tämän työn tilannut toimeksiantaja, teki työtä DIGILE-hankkeen hallinnoimalle FORGE Service Development Lab palvelulle. Kyseinen palvelu oli käytännössä pilvipalvelu, jota markkinoitiin suomalaisille yrityksille. FORGEN tarkoitus oli tukea sille asiakkaaksi tulevia yrityksiä kehittämään omia Internet-palveluita tarjoamalla niille toimintaympäristön jossa kehittää kyseiset palvelut, ja asiantuntijoita kehitystyön tueksi. Projekti tähtäsi siihen, että suomalaisyritysten parissa olevan tiedon ja taidon määrä, joka liittyy Internet-palveluiden kehittämiseen, kasvaa FORGEN myötä. (DIGILE, 2015)

FORGE Service Development Labin ytimessä oli avoimen lähdekoodin virtualisointiohjelmisto OpenStack, joka on ohjelmoitu Python-kielellä. Kainuun Edun projektissa mukana oleva henkilöstö oli osallisena asiakaspalvelussa ja teknisessä tuessa, jossa työtehtävät keskittyivät FORGEN ytimessä olevan OpenStack-ohjelmiston ja sen parissa olevien resurssien hallintaan. Tätä tarkoitusta varten Kainuun Edulla oli ilmennyt tarve tutkia työvälineitä, joilla kykenee hallitsemaan virtualisointiohjelmita kuten OpenStackia, ja mahdollisia muita vastaavia ohjelmia, jos FORGEN pariin olisi otettu vaihtoehtoja OpenStackin rinnalle.

Varteenotettavaksi ja ainoaksi testattavaksi vaihtoehdoksi löytyi Apachen ohjelmistosäätiön kehittämä Python-ohjelmointikieltä käyttävä LibCloud-komentokirjasto. Tällä komentokirjastolla oli ilmoitettu yhteensopivuus OpenStackin kanssa, eikä sen asentaminen olisi häirinnyt itse OpenStackin toimintaa, toisin kuin muut löydetty vaihtoehdot.

LibCloud voidaan asentaa erilleen itse OpenStackista, omaksi kerroksekseen virtualisointiohjelmistojen yläpuolelle, josta komentoja voidaan antaa useammille ohjelmistoille. Seuraavan sivun kuva demonstroi tätä järjestelyä.



Kuva 3. Havainne LibCloudin sijainnista useiden virtualisointiohjelmistojen hierarkiassa. LibCloud tarjoaa yhtenäisen käyttöliittymän, jonka ansiosta useiden eroavien virtualisointiohjelmistojen käyttö, muokkaus ja käsitytys onnistuvat yhden ohjelman kautta.

Apachen ohjelmistosäätiön projektihautoimosta löytyi useita muita potentiaalisia kilpailijoita LibCloudille, kuten Deltacloud, Whirr, Tashi ja Nuvem. Nämä ohjelmistot kuitenkin eivät olleet varteenotettavia kandidaatteja useista erisyistä.

Deltacloud oli joukoista kaikkein tuoreimmalla julkaisulla, mutta sen kehitystyö lopetettiin kesäkuussa ennen opinnäytetyön aloitusta, mikä sulki sen pois testauskandidaateista (Apache Software Foundation, 2015). Whirr-projekti oli lopetettu aiemmin maaliskuussa (Apache Software Foundation, 2015). Tashi-projekti oli jo hylätty vuonna 2012 (The Apache Software Foundation, 2015), ja Nuvemista ei ole saatavilla mitään julkaisua tai edes kotisivuja (The Apache Software Foundation, 2015).

Ainoa kunnollinen vaihtoehto oli OCCi tai Open Cloud Computing Interface, mutta sitä ei valittu testattavaksi. Toisin kuin LibCloud, OCCi olisi asentunut suoraan osaksi OpenStackin infrastruktuuria. Sen asennus sekä poisto olisivat voineet haitata OpenStackin toimintaa. Kainuun Etu täten ilmoitti, ettei OCCiä saisi testata FORGEN ympäristössä. Vaihtoehtona olisi ollut rakentaa

testikäyttöä varten toinen OpenStack-ympäristö, mutta sen hankaluuden ja rajoitetun ajan takia tähän ei päädytty.

Täten työkuvaiksi oli muotoutumassa LibCloud-komentokirjaston asennus ja sen toiminnallisuuden testaaminen mustalaatikko-testaukseksi kutsutulla metodilla. Mustalaatikko-testauksessa testaavalla henkilöllä ei ole pääsyä näkemään mitä ohjelman sisällä tapahtuu testauksen aikana, vaan arviointikriteereinä ovat vain syötetty tieto ja siitä saadut tulokset. Harmaalaatikko- ja valkolaatikko-metodit olisivat periaatteessa olleet mahdollisia, LibCloudin lähdekoodin ollessa avointa, mutta ajan ja asiantuntemuksen puutteen takia tarkkaa ohjelmiston käyttäytymistä kooditasolla ei voitu tarkkailla. (TutorialsPoint, 2015)

LibCloudin kohdalla tämä merkitsi kaikkien komentojen testausta, joilla oli edellytys toimia LibCloudin oman dokumentaation perusteella. Testaukseen valikoituneet käskyt syötettäisiin ohjelmalle ja niiden toteutumista seurattaisiin OpenStackin puolelta sekä LibCloudin omasta terminaalista virheiden ja lopputuloksen toteutukseksi. Lopuksi tämä prosessi dokumentoitaisiin osaksi tätä opinnäytetyötä. Kyseisen komentokirjaston dokumentaatiosta paljastui, että saatavilla olleen version tuki OpenStackille ei ollut täydellinen. Osa komennoista ei ollut loppuun saakka kirjoitettuja, tai niiden käyttöön vaadittuja tietoja ei voitu tuottaa LibCloudin saatavilla olleella versiolla.

Työympäristönä palveli FORGEN sisälle pystytetty virtuaalipalvelin, jolle LibCloud-komentokirjasto asennettiin sen vaatimien ohjelmien kanssa. Tältä palvelimelta käsin sitten käskytettiin FORGEN OpenStack-ohjelmistoa ja seurattiin OpenStackin oman käyttöliittymän kautta, miten annetut käskyt toteutuivat.

4 OHJELMISTO JA SEN RAJOITTEET

Apache LibCloud näki ensijulkaisunsa pelkkänä LibCloudina vuonna 2009, ja se liittyi Apache-ohjelmistosäätiön projektihautomoon pian julkaisunsa jälkeen samana vuonna. Vuodesta 2011 eteenpäin kehitystyö on jatkunut yhtenä hautomon ensisijaisista projekteista. Nykyinen ja opinnäytetyöhön käytetty ohjelmistoversio on 0.18.0, joka julkaistiin elokuussa 2015. (The Apache Software Foundation, 2015)

Käskykirjaston ohjelmointikieli on Python. LibCloud tukee Python-kielen versioita 2.5, 2.6, 2.7, 3.0 ja PyPyä, joka on vaihtoehtoinen toteutus Python-kielen kirjastoista ja komennoista.

Ohjelmiston omassa dokumentaatiossa ilmoitetaan, että tuki kaikille toiminnoille OpenStack-ohjelmistossa ei ole vielä täydellinen. Virtuaalikoneiden käsittelyssä LibCloud ei kykene luomaan tai käynnistämään virtuaalikoneita. Tallennustilan käsittelyssä LibCloud ei voi vielä luoda tai ottaa listaa virtuaalikoneista otetuista snapshot-tallennuskuvista. Salausavaintietojen käsittelylle ei ole ollenkaan tukea OpenStackille vielä. Sisällön jakamisverkostoissa (CDN) OpenStack Swift-ajuri ei tue vielä objektien sallimista. (The Apache Software Foundation, 2015)

LibCloudin dokumentaatiossa esiintyviä ohjelmakäskyjä, jotka liittyvät edellä mainittujen ominaisuuksien käyttöön, joita ei ole vielä kehitetty valmiiksi, on siten sivuutettu testausvaiheessa.

5 OHJELMISTON ASENNUS JA KÄYTTÖ

Suosittelua ja työssä käytetty toimintaympäristö on Linux-pohjainen palvelin terminaalinäkökulmalla. Graafiset Linux-distribuutiot, Windows-käyttöjärjestelmät ja muut vaihtoehdot eivät kuitenkaan ole poissuljettuja, jos niillä saa ajettua Python-ohjelmointikielellä toimivaa ohjelmistoa.

Apache LibCloudin asennusta varten FORGE-ympäristöön luotiin Ubuntu 14.04.03 Trusty Tahr palvelin terminaalinäkökulmalla. On suositeltavaa asentaa päivitykset käskyillä `'sudo apt-get update'` ja `'sudo apt-get upgrade'`, jotta palvelimen ohjelmistot ovat ajan tasalla ennen LibCloudin asennusta.

LibCloud itse asentuu Python-Pip-ohjelman kautta, joten tämä tulee ensin asentaa palvelimelle komennolla `'sudo apt-get install python-pip'`. Kun Python-Pip on asennettu, voidaan sen kautta sitten asentaa LibCloud komennolla `'sudo pip install apache-libcloud'`.

Kun nämä on asennettu, on hyvä tarkastaa ensin Pythonin toimivuus palvelimella, mikä onnistuu pelkällä `'python'`-komennolla Linuxin konsoliin. Jos Python-kielinen käyttöliittymä käynnistyy, palvelin ja LibCloud ovat valmiita toimintaan.

Opinnäytetyössä käytetty keino välittää komentoja LibCloudille ja sitä kautta OpenStackiin on ollut kirjoittaa Linuxissa skripti-tiedostoja, joiden sisälle Python-kieliset käskyt ovat sijoitettu. Kaikki käskyt jotka ovat Python-kielellä kirjoitettuja, tulee olla `python -c " " "`-komennon sisään pakattuja, sitaattimerkkien välissä.

Jotta LibCloud kykenisi viestimään OpenStackin kanssa, sille tulee osoittaa rajapinta tai API, jonne LibCloudin viestit suuntautuvat. Halutut rajapinnat löytyvät OpenStackin puolelta Compute-näkymästä hakemalla Access&Security-kohdan ja sieltä API Access välilehden, kuten seuraavalla sivulla oleva kuva demonstroi.

Project

Compute

Overview

Instances

Volumes

Images

Access & Security

Network

Object Store

Admin

Access & Security

Security Groups
Key Pairs
Floating IPs
API Access

API Endpoints

Service	Service Endpoint
Compute	https://cloud.forgeservicelab.fi:8777/v2/kainuunetu
Network	https://cloud.forgeservicelab.fi:9697/
Volumev2	https://cloud.forgeservicelab.fi:8780/v2/kainuunetu
Computev3	https://cloud.forgeservicelab.fi:8777/v3
S3	https://cloud.forgeservicelab.fi:8081
Image	https://cloud.forgeservicelab.fi:9293
Volume	https://cloud.forgeservicelab.fi:8780/v1/kainuunetu
EC2	https://cloud.forgeservicelab.fi:8772/services/Cloud
Object Store	https://cloud.forgeservicelab.fi:8081/v1/AUTH_kainuunetu
Identity_V3	https://cloud.forgeservicelab.fi:5001/v3
Identity	https://cloud.forgeservicelab.fi:5001/v2.0

Displaying 11 items

Kuva 4. Havainne API-rajapintojen sijainnista OpenStackin verkkoselaimessa olevassa käyttöliittymässä.

Autentikaatioskriptin, joka sisältää käyttäjätunnuksen ja rajapinnan, tulisi siten näyttää tältä Kainuun Edun käyttämän projektin osoitteeseen ohjattuna, Nova-virtuaalikonepalvelulle ohjattuna:

```
python -c "
from libcloud.compute.types import Provider
from libcloud.compute.providers import get_driver

import libcloud.security

# This assumes you don't have SSL set up.
# Note: Code like this poses a security risk (MITM attack) and
# that's the reason why you should never use it for anything else
# besides testing. You have been warned.
#libcloud.security.VERIFY_SSL_CERT = False

OpenStack = get_driver(Provider.OPENSTACK)
driver = OpenStack('käyttäjätili',
    'tilin salasana',
    ex_force_auth_version='2.0_password',
    ex_force_auth_url='https://cloud.forgeservicelab.fi:5001',
    ex_tenant_name='kainuunetu',
    ex_force_service_name='nova',
    ex_force_service_region='KajaaniDC1-prod') "
```

Edellä esitettyyn viestiin on sijoitettu tiedot, jotta LibCloudin käyttäjän komennot ohjautuvat virtuaalikoneita käsittelevään Nova-palveluun. Jos halutaan komentaa tiedontallennusta hoitavaa Swift-palvelua, silloin autentikaatioviestistä tulee lisätä ja vaihtaa seuraavat kohdat:

```
from libcloud.storage.types import Provider
from libcloud.storage.providers import get_driver

OpenStack = get_driver(Provider.OPENSTACK_SWIFT)
ex_force_auth_url='https://cloud.forgeservicelab.fi:5000',
ex_force_service_type='object-store',
ex_force_service_name='swift'
```

Autentikaatioviesti voi toimia viestipohjana, jossa halutut käskyt kirjoitetaan autentikaatiotietojen jälkeen suoritettaviksi. Kehittyneemmät Python-kielen käyttäjät voivat myös luoda oman Python-kielisen tiedoston autentikaatiotiedoista, joita voidaan kutsua skripti-tiedostossa muita käskyjä suorittaessa. Opinnäytetyössä komentojen testit suoritettiin alkeellisemmalla autentikaatioviestipohja-metodilla, jotta aikaa ei olisi kulunut Python-kielen opetteluun erikseen. Tästä on huomioitavaa, että tuotantokäytössä tämä on erittäin huono ja turvaton tapa jota ei tulisi käyttää, koska jokaisessa komennossa on tärkeitä käyttäjätunnustietoja näkyvissä. Tietoturvan ylläpitämiseksi skripti-tiedostot tulisi olla tarpeeksi korkeiden käyttöoikeuksien takana, ettei kuka tahansa pääse lukemaan tai suorittamaan niitä.

6 OHJELMAKÄSKYT

Seuraavaksi esiteltävät ohjelmakäskyt ovat itse LibCloud-komentokirjaston anti ja sisältö. Niillä komennetaan OpenStack-ohjelmistoa suorittamaan halutut toimet. Ne tullaan esittämään muodossa, jossa vahvennettu teksti on itse käsky, jollaisena se tulee esiintyä toimiakseen osana skripti-tiedostoa. Vahvennettua tekstiä seuraa selite joka kertoo käskyn tarkoituksen ja muut huomioitavat asiat. Viimeisenä selitetään käskyssä esiintyvät muuttujat.

6.1 Tietoja hakevat käskyt

Nämä käskyt hakevat ja muuttavat OpenStackista saatavaa tietoa muotoon, jossa LibCloud ymmärtää niitä. Toiminnalliset käskyt yleensä vaativat muuttujia ja tietoja, jotka on ensin käsitelty näillä käskyillä.

ex_get_node_details(node_id)

Hakee palvelimen tunnistetiedot ja muuttaa ne Node-tietueeksi muita käskyjä varten.

node_id (str) – Kohdepalvelimen ID.

ex_get_node_security_groups(node)

Palauttaa listan sääntöryhmistä joissa kohdepalvelin on jäsenenä. Ei toiminnallinen.

ex_get_size(size_id)

Muuttaa palvelimen flavour-tietueen muotoon jota LibCloud voi tulkita. Käytetään palvelimien flavour-tietuetta vaihtavissa komennoissa.

size_id (str) – Flavourin ID

get_image(image_id)

Hakee asennuskuvan sen ID:n perusteella. Muuttaa tietueen NodeImage-muotoon. Käytetään asennuskuvia käsittelevien komentojen yhteydessä.

image_id (str) – Halutun asennuskuvan ID.

ex_get_console_output(node, length=None)

Hakee palvelimen konsolin viimeisimmät rivit. Todettu toimimattomaksi.

node (Node) – Kohdepalvelin

length (int) – Konsolista otettavien rivien määrä.

ex_get_floating_ip(ip)

Hakee kelluvan IP-osoitteen ja muuttaa sen OpenStack_1_1_FloatingIpAddress-tietueeksi. Käskey palauttaa virheilmoituksen, eikä siten ole käytettävissä.

ip (str) – Haluttu IP-osoite

ex_get_metadata(node)

Hakee palvelimen avain- ja arvometadatan. Käskey ei ole vielä viimeistelty.

node (Node) – Kohdepalvelin

ex_get_metadata_for_node(node)

Hakee palvelimeen liittyvät metadatat. Käskey ei ole vielä viimeistelty.

node (Node) – Kohdepalvelin

ex_list_floating_ip_pools()

Listaa vapaat kelluvat IP-osoitepoolit. Käskey paljastui vielä toimimattomaksi.

ex_list_floating_ips()

Listaa vapaat kelluvat IP-osoitteet. Käskey paljastui vielä toimimattomaksi.

ex_list_networks()

Listaa käytettävissä olevat aliverkot.

ex_list_security_groups()

Listaa käytettävissä olevat sääntöryhmät. Ei toimi.

list_images(location=None, ex_only_active=True)

Listaa kaikki aktiiviset asennuskuvat.

`ex_only_active` - Määrää listataanko kaikki vai vain käytettävissä olevat.

`list_nodes(ex_all_tenants=False)`

Listaa kaikki palvelimet toimialueesta tai tenantista.

`ex_all_tenants` (bool) – Listaa kaikkien saatavilla olevien toimialueiden palvelimet. Vaatii admin-oikeudet OpenStackiin toimiakseen.

`openstack_connection_kwargs()`

Toistaa käytössä olevat OpenStackin autentikaatitiedot.

6.2 Toiminnalliset käskyt

Toiminnalliset käskyt suorittavat jonkin toiminnon OpenStackissa. Tällaisia toimintoja ovat esimerkiksi virtuaalikoneiden käsittely ja muokkaus tai niitä säätelevien sääntöryhmien luonti.

`copy_image(source_region, node_image, name, description=None)`

Kopioi asennuskuvan toisesta OpenStack-toimialueesta nykyiseen. Käsky ei toiminut.

`source_region` (str) – kohdetoimialue

`node_image` (NodeImage:) – kohteena olevan asennuskuvan nimi

`name` (str) – uusi nimi asennuskuvalla nykyiseen toimialueeseen

`description` – mahdollinen kuvaus

`create_image(node, name, metadata=None)`

Luo olemassa olevasta palvelimesta asennuskuvan. Autentikaatiovaiheessa syötetyllä OpenStack-tilillä tulee olla admin-oikeudet, koska asennuskuvasta tehdään kaikille julkinen, tai toisin sanoen pitää olla käyttöoikeus luoda public-alueen asennuskuvia.

node (Node) – Kohdepalvelin

name (str) – Uuden asennuskuvan nimi

metadata (dict) – Mahdollinen avainpari asennuskuvalle (ei toimi vielä)

create_volume(size, name, location=None, snapshot=None, ex_volume_type=None)

Luo kovalevyvolumin.

size (int) – Volumin koko gigatavuissa ilmaistuna

name (str) – Volumin nimi

location (NodeLocation) – Mihin sijaintiin voluumi luodaan. Voidaan jättää määrittelemättä.

snapshot (VolumeSnapshot) – OPTIONAALINEN: Asennuskuva jonka pohjalta voluumi luodaan.

ex_volume_type (str) – OPTIONAALINEN: Volumin tyyppi.

Palauttaa uuden volumin muodossa StorageVolume.

delete_image(image)

Poistaa nimetyn asennuskuvan. Kuvan tieto tulee hakea erikseen get_image-komennolla.

ex_attach_floating_ip_to_node(node, ip)

Liittää vapaana olevan kelluvan IP-osoitteen palvelimeen.

node (Node) – palvelintietue johon osoite liitetään

ip (str tai OpenStack_1_1_FloatingIpAddress) – Vapaa kelluva IP-osoite. Voi olla get-komennolla haettu tai suoraan tekstinä annettu.

ex_confirm_resize(node)

Vahvistaa palvelimen flavourin muutos.

ex_create_floating_ip()

Luo uuden kelluvan IP-osoitteen. Käsky ei toiminut.

ex_create_network(name, cidr)

Luo uusi verkko. Käsky välillä suoriutui onnistuneesti, välillä palautti virheilmoituksen.

name (str) – Uuden verkon nimi.

cidr (str) – Uuden verkon cidr, tai verkkoavaruus.

ex_create_security_group(name, description)

Luo uuden sääntöryhmän.

name (str) – Sääntöryhmän nimi.

description (str) – Sääntöryhmän kuvaus.

ex_create_security_group_rule(security_group, ip_protocol, from_port, to_port, cidr=None, source_security_group=None)

Luo uuden säännön sääntöryhmälle. Alustana voi käyttää olemassa olevaa ryhmää. Käsky ei toiminut.

security_group (OpenStackSecurityGroup) – Sääntöryhmä johon sääntö tulee

ip_protocol (str) – Protokolla (TCP, UDP jne.) joka sääntö koskee.

from_port (int) – Portti josta aloittaa sääntö.

to_port (int) – Portti johon lopettaa sääntö.

cidr (str) – CIDR-notaatio lähde-IP:tä koskien

source_security_group (L{OpenStackSecurityGroup}) – Vaihtoehtoisesti olemassa oleva sääntöryhmä CIDRin sijasta.

ex_delete_floating_ip(ip)

Poistaa kelluvan IP-osoitteen. Käskey palauttaa virheilmoituksen.

ex_delete_network(network)

Poistaa aliverkon. Ei käytettävissä.

ex_delete_security_group(security_group)

Poistaa sääntöryhmän. Ei käytettävissä.

ex_delete_security_group_rule(rule)

Poistaa säännön. Ei käytettävissä.

ex_detach_floating_ip_from_node(node, ip)

Irroittaa kelluvan IP-osoitteen palvelimesta.

node (Node) – Kohdepalvelin

ip (str tai OpenStack_1_1_FloatingIpAddress) – poistettava osoite

ex_resize(node, size)

Vaihtaa palvelimen flavouria. Toimintakuntoinen, mutta käskey voi epäonnistua OpenStackin omien virheiden takia.

node (Node) – Kohdepalvelin

size (NodeSize) – Uusi flavour (hae vastaavalla ex_get_size-komennolla)

ex_revert_resize(node)

Nollaa odottavan flavourin vaihdon palvelimelta.

node (Node) – Kohdepalvelin

ex_set_metadata(node, metadata)

Asettaa palvelimelle metadatat. Käskeyn toimivuutta ei voida varmentaa muita käskeyjä koskevien puutteiden takia.

node (Node) – Kohdepalvelin

metadata (dict) – Avain/metadatum joka liittyy palvelimeen.

ex_set_server_name(node, name)

Muuttaa palvelimen nimen.

node (Node) – Kohdepalvelin

name (str) – Palvelimen uusi nimi.

ex_soft_reboot_node(node)

Suorittaa pehmeän uudelleenkäynnistyksen palvelimelle.

node (Node) – Kohdepalvelin

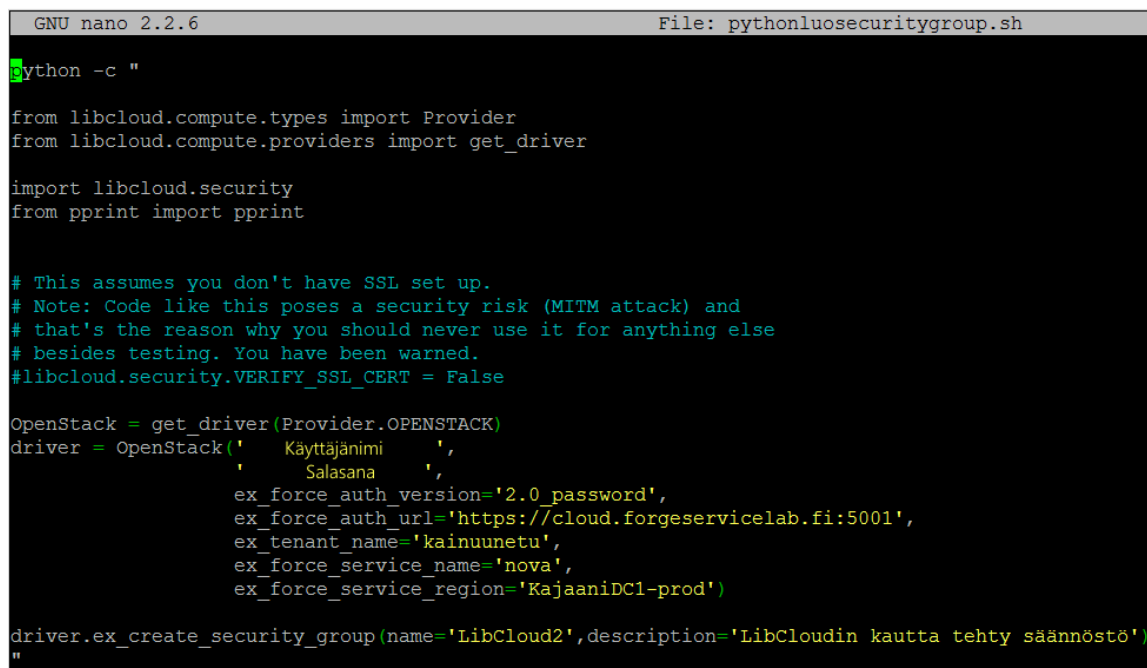
ex_hard_reboot_node(node)

Pakottaa kovan uudelleenkäynnistyksen palvelimelle.

node (Node) – Kohdepalvelin

6.3 Esimerkkejä komennoista osana skriptejä

Alla olevan kuvan skriptissä varsinainen komento on viimeinen rivi pohjalla, jossa `ex_create_security_group`-komento luo uuden turvallisuussääntöryhmän jonka nimeksi on annettu 'LibCloud2'. Sille on myös annettu kuvaus, jonka avulla voidaan välittää lisätietoa ryhmän tarkoituksesta ja luonteesta.



```
GNU nano 2.2.6                                File: pythonluosecuritygroup.sh

python -c "

from libcloud.compute.types import Provider
from libcloud.compute.providers import get_driver

import libcloud.security
from pprint import pprint

# This assumes you don't have SSL set up.
# Note: Code like this poses a security risk (MITM attack) and
# that's the reason why you should never use it for anything else
# besides testing. You have been warned.
#libcloud.security.VERIFY_SSL_CERT = False

OpenStack = get_driver(Provider.OPENSTACK)
driver = OpenStack('    Käyttäjänimi    ',
                  '    Salasana    ',
                  ex_force_auth_version='2.0_password',
                  ex_force_auth_url='https://cloud.forgeservicefab.fi:5001',
                  ex_tenant_name='kainuunetu',
                  ex_force_service_name='nova',
                  ex_force_service_region='KajaaniDC1-prod')

driver.ex_create_security_group(name='LibCloud2',description='LibCloudin kautta tehty säännöstö')
"
```

Kuva 5. Esimerkki skriptistä, joka luo uuden sääntöryhmän OpenStackiin.

Käskyn edessä oleva 'driver'-muuttuja on saatu ensin hakemalla OpenStackin 'get_driver'-muuttuja, joka on nimetty sitten 'OpenStack'-muuttujaksi. Sitten tälle 'OpenStack'-muuttujalle on annettu 'get_driver'-käskyn vaatimat tiedot, tai toisin sanoen autentikaatioskripti. Näin on saadaan 'driver'-muuttuja joka sisältää autentikaatitiedot. Osaavampi Python-ohjelmoija kykenee tekemään pysyvän tiedoston näistä tiedoista, jolloin jokaisen skriptin alussa ei tarvitsisi toistaa autentikaatiota.

Alla olevan kuvan skriptissä muuttuja 'kohdenode' saa arvonsa komennosta `ex_get_node_details`, joka hakee virtuaalipalvelimen jonka ID on nähtävissä kuvassa. Tämän komennon alla on seuraavaksi kolme eri komentoa, jotka hakevat salausavaimiin liittyviä tietoja 'kohdenode'-muuttujan palvelimesta ja sen konsolin kymmenen viimeistä riviä. Alimmaisena on komento joka on estetty suoriutumasta, jolla irroitetaan 'kohdenode'-muuttujan palvelimesta kuvassa näkyvä IP-osoite.

```
GNU nano 2.2.6 File: pythonnoodit.sh

python -c "

from libcloud.compute.types import Provider
from libcloud.compute.providers import get_driver

import libcloud.security
from pprint import pprint

# This assumes you don't have SSL set up.
# Note: Code like this poses a security risk (MITM attack) and
# that's the reason why you should never use it for anything else
# besides testing. You have been warned.
#libcloud.security.VERIFY_SSL_CERT = False

OpenStack = get_driver(Provider.OPENSTACK)
driver = OpenStack('    Käyttäjänimi    ',
                  '    Salasana    ',
                  ex_force_auth_version='2.0_password',
                  ex_force_auth_url='https://cloud.forgeservicelab.fi:5001',
                  ex_tenant_name='kainuunetu',
                  ex_force_service_name='nova',
                  ex_force_service_region='KajaaniDC1-prod')

kohdenode=driver.ex_get_node_details('3259a761-03a4-407f-aea3-b0a6cea0e88d')
pprint(driver.ex_get_metadata(kohdenode))
pprint(driver.ex_get_metadata_for_node(kohdenode))
pprint(driver.ex_get_console_output(kohdenode,length=10))
#driver.ex_detach_floating_ip_from_node(kohdenode,'193.166.24.106')
"
```

Kuva 6. Esimerkki skriptistä, joka muuttaa palvelimen tiedot LibCloudin lukemaan muotoon ja sitten hakee tietoja kyseisestä palvelimesta.

6.4 Virheet ja havaitut puutteet

LibCloudin testauksen aikana ilmeni virheitä ja puutteita, joista osa oli käyttäjän korjattavissa, mutta valtaosa oli sellaisia puutteita joita olivat käskykirjaston kehittäjien vastuulla korjattaviksi.

ex_get_node_details(node_id)

Muut käskykirjaston käskyt vaativat tietonsa hankittavaksi kirjastoon sisälletyillä keinoilla, mutta tämä oletamus ei päde tämän käskyn kohdalla. *Ex_get_node_details*-käskyn vaatima tietue 'node_id' vastaa OpenStackin verkkoselain-käyttöliittymästä nähtyä palvelimen ID:tä, eikä LibCloudin komennoilla nähtävää ID:tä.

ex_get_node_security_groups(node)

LibCloudin dokumentaatio ei maininnut käskyn olevan kesken, mutta käytön aikana paljastui, ettei käsky palauta mitään tietoa käyttäjälle. Käsky on täten kesken tai rikki.

ex_get_console_output(node, length=None)

LibCloudin dokumentaatiossa ei ollut mainintaa käskyn toimimattomuudesta, mutta käytön aikana paljastui, ettei käsky palauta mitään tietoa käyttäjälle. Käsky on täten kesken tai rikki.

ex_get_floating_ip(ip)

LibCloudin dokumentaatiossa ei ollut mainintaa käskyn toimimattomuudesta, mutta käytön aikana komento palautti virheilmoituksen joka ilmoitti käskyn toteutuksen olevan kesken eikä siten olevan käytettävissä.

ex_get_metadata(node)

Käsky palautti vain nollan käyttäjälle. Käskyn toteutus on ilmeisesti kesken.

ex_get_metadata_for_node(node)

LibCloudin dokumentaatio perusteella käsky ei ollut käytettävissä. Testaus paljasti tiedon paikkansa pitäväksi käskyn keskeneräisyyden takia.

ex_list_floating_ip_pools()

Käsky palautti virheilmoituksen, joka ilmoitti että sitä oltu vielä viimeistelty käyttöön.

ex_list_floating_ips()

Käsky palautti virheilmoituksen, joka ilmoitti että sitä oltu vielä viimeistelty käyttöön.

ex_list_security_groups()

Käsky palautti virheilmoituksen, joka ilmoitti että sitä oltu vielä viimeistelty käyttöön.

copy_image(source_region, node_image, name, description=None)

Käsky palautti virheilmoituksen, joka ilmoitti että sitä oltu vielä viimeistelty käyttöön.

ex_create_floating_ip()

Käsky ei onnistunut löytämään mitään julkisia resursseja OpenStackista jonne luoda uutta IP-osoitetta, ja palautti virheilmoituksen. Käsky oli suoritettu korotetuilla käyttäjäoikeuksilla. Virhe ilmeisesti on käskyn koodissa.

ex_create_security_group_rule(security_group, ip_protocol, from_port, to_port, cidr=None, source_security_group=None)

Käsky ei ole käytettävissä, koska se vaatii tietoja joita ei voi hakea. Tähän tarkoitukseen tehdyt käskyt eivät ole vielä implementoitu.

ex_delete_floating_ip(ip)

Käsky palautti vain virheilmoituksen useista yrityksistä huolimatta. Ilmeistä syytä epäonnistumiselle ei voida osoittaa.

ex_create_network(name, cidr)

Käsky välillä palautti virheilmoituksen, jonka syytä ei voitu osoittaa.

ex_delete_network(network)

Ei käytettävissä, koska käskykirjastossa ei ole vielä keinoa yksilöidä kohde-aliverkkoa jota poistaa.

ex_delete_security_group(security_group)

Ei käytettävissä, koska käskykirjastossa ei ole vielä keinoa yksilöidä kohdesääntöryhmää jota poistaa.

ex_delete_security_group_rule(rule)

Ei käytettävissä, koska käskykirjastossa ei ole vielä keinoa yksilöidä kohdesääntöä halutusta sääntöryhmästä jota poistaa.

ex_set_metadata(node, metadata)

Käskyn toimivuutta ei voida varmentaa muita käskyjä koskevien puutteiden takia.

Autentikaatio epäonnistuu

Jos autentikointiyritys palauttaa virhetekstin:

libcloud.common.exceptions.BaseHTTPError: {"error": {"message": "get_version_v3() got an unexpected keyword argument 'auth'", "code": 400, "title": "Bad Request"}}

Kyseessä on huolimattomasti kirjoitetusta API-rajapinnan osoitteesta. Osoitteella (esim. <https://cloud.forgeservicelab.fi:5001>) ei tule joko olla liitettä sen lopussa, tai liitteen tulee olla muotoa '/v2/tokens'. Pelkkä '/v2' aiheuttaa virheen.

7 SELVITYKSEN YHTEENVETO

Ottaen huomioon valmiiksi esitetyt puutteet Apache LibCloudin omassa dokumentaatiossa, ja opinnäytetyön aikana havaitut ylimääräiset puutteellisuudet, tämän selvitystyön mielipiteeksi muodostuva kanta ei tue Apache LibCloudin käyttöä osana FORGE Service Development Lab-hanketta.

Ne ohjelmakäskyt, jotka toimivat OpenStack-ympäristön kanssa, ovat rajallisia ja ovat luonteeltaan olemassa olevan sisällön hallinnointia varten tarkoitettuja. Tähän tarkoitukseen OpenStackin jo olemassa olevat käyttöliittymät tyydyttävät tarpeet paremmin. Apache LibCloudin kautta ei kykene tekemään uusia virtuaalipalvelimia, mikä on yksi tärkeimmistä asioista, joita uudet asiakkaat tarvitsisivat. Palvelimiin ei voi myöskään liittää käyttöturvallisuuden kannalta olennaisia salausavaimia.

Lisäksi eräät olennaiset tiedot hakevat käskyt, jotka ovat olennaisia toisten käskyjen käyttöön, puuttuvat ilman mainintaa Apache LibCloudin omassa dokumentaatiossa tästä puutteesta.

Apache LibCloud täten ei sovellu nykyisessä versiossaan käytettäväksi ensisijaisena hallintatyökaluna OpenStack-ympäristöön. Tulevissa versioissa puutteet saattavat olla korjattuja, mutta se vaatii odottamista.

Mikäli LibCloudin käyttöönotto olisi välttämätöntä, sitä varten tulisi varata Python-ohjelmointikieltä osaavia ohjelmoijia, jotka kykenevät paikkaamaan komentokirjastossa esiintyviä vaillinaisuuksia. Tällä toimella on myös omat kustannuksensa, jotka vielä entisestään lisäävät kannusteita olla ottamatta ohjelmistoa käyttöön nykyisien toimitapojen rinnalle.

Huomioiden vielä se fakta, että FORGE Service Development Labin liiketoiminta lopetettiin 16.11.2015, enempiä kehitysehdotuksia ei voida esittää hankkeelle.

LÄHDELUETTELO

Apache Software Foundation. (haettu 8. 11 2015). *Apache Deltacloud*. Noudettu osoitteesta <http://attic.apache.org/projects/deltacloud.html>

Apache Software Foundation. (haettu 8. 11 2015). *Etusivu*. Noudettu osoitteesta <https://whirr.apache.org/>

Bunn, F.;Simpson, N.;Peglar, R.;& Nagle, G. (13. 10 2003). Storage Virtualization. *SNIA Technical Tutorial*, 3-5. Storage Networking Industry Association. Noudettu osoitteesta SNIA Technical Tutorial: <http://www.snia.org/sites/default/files/sniavirt.pdf>

BusinessWire. (19. 9 2012). *OpenStack Launches as Independent Foundation, Begins Work Protecting, Empowering and Promoting OpenStack*. Noudettu osoitteesta

<http://www.businesswire.com/news/home/20120919005997/en/OpenStack-Launches-Independent-Foundation-Begins-Work-Protecting>

Conroy, S. P. (11. 8 2011). *History of virtualization*. Noudettu osoitteesta <http://www.everythingvm.com/content/history-virtualization>

DIGILE. (haettu 5. 11 2015). *Tietoa meistä*. Noudettu osoitteesta FORGE Service Labin sivusto: <https://forgeservicelab.fi/fi>

Foundation, O. (haettu 18. 11 2015). *What is OpenStack?* Noudettu osoitteesta www.openstack.org

Interoute Communications Limited. (haettu 17. 11 2015). *What is Cloud Computing?* Noudettu osoitteesta <http://www.interoute.com/cloud-article/what-cloud-computing>

Interoute Communications Limited. (haettu 17. 11 2015). *What is IaaS?* Noudettu osoitteesta <http://www.interoute.com/what-iaas>

Interoute Communications Limited. (haettu 17. 11 2015). *What is PaaS?* Noudettu osoitteesta <http://www.interoute.com/what-paas>

Interoute Communications Limited. (haettu 17. 11 2015). *What is SaaS?* Noudettu osoitteesta <http://www.interoute.com/what-saas>

Marko, K. (20. 11 2014). *Open Source Vs. Proprietary: Time For A New Manifesto*. Noudettu osoitteesta Network Computing-sivusto, UBM Tech: http://www.networkcomputing.com/data-centers/open-source-vs-proprietary-time-for-a-new-manifesto/d/d-id/1317625?page_number=2

Moorman, L. (18. 7 2010). *Opening the Rackspace cloud: Rackspace Blog*. Noudettu osoitteesta <http://blog.rackspace.com/opening-the-rackspace-cloud/>

OnlyJob. (22. 3 2012). *Perl, Python, Ruby, PHP, C, C++, Lua, tcl, javascript and Java comparison*. Noudettu osoitteesta <http://raid6.com.au/~onlyjob/posts/arena/>

Python Software Foundation. (haettu 8. 11 2015). *Applications for Python*. Noudettu osoitteesta <https://www.python.org/about/apps/>

Red Hat Corporation. (haettu 11. 11 2015). *What is Virtualization?* Noudettu osoitteesta http://www.redhat.com/f/pdf/virtualization/gunner_virtual_paper2.pdf

Rossum, G. v. (13. 1 2009). *Introduction and overview*. Noudettu osoitteesta <http://python-history.blogspot.fi/2009/01/introduction-and-overview.html>

Rossum, G. v. (20. 1 2009). *Python History*. Noudettu osoitteesta <http://python-history.blogspot.fi/2009/01/brief-timeline-of-python.html>

SDxCentral. (haettu 12. 11 2015). *What is Network Virtualization*. Noudettu osoitteesta <https://www.sdxcentral.com/resources/network-virtualization/whats-network-virtualization/>

The Apache Software Foundation. (haettu 8. 11 2015). *Apache Tashi Project Incubator*. Noudettu osoitteesta <http://incubator.apache.org/projects/tashi.html>

The Apache Software Foundation. (haettu 8. 11 2015). *Project Nuvem Apache Incubator*. Noudettu osoitteesta <http://incubator.apache.org/projects/nuvem.html>

The Apache Software Foundation. (haettu 29. 10 2015). *Tietoja meistä: LibCloud Apache-projekti*. Noudettu osoitteesta <https://libcloud.apache.org/about.html>

The Apache Software Foundation. (haettu 29. 10 2015). *Tuetut palveluntarjoajat*. Noudettu osoitteesta https://libcloud.readthedocs.org/en/latest/supported_providers.html

TIOBE Software BV. (haettu 16. 11 2015). *TIOBE Index for November 2015*. Noudettu osoitteesta <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

TutorialsPoint. (haettu 11. 19 2015). *Software Testing - Methods*. Noudettu osoitteesta http://www.tutorialspoint.com/software_testing/software_testing_methods.htm